Matrix Multiplication With Fixed Matrices and Polynomial Evaluation With Fixed Polynomials

J. E. Savage[1]
Communications Research Section

Others have shown that the conventional method for computing $m \times n$ matrix-vector products and Horner's rule for evaluating polynomials are optimal when matrix and vector elements as well as polynomial coefficients and polynomial variables are indeterminate. In this article, the calculation of matrix-vector products and the evaluation of polynomials are treated when the matrix elements and polynomial coefficients are known and drawn from a set of size s. It is shown that the algorithms which are optimal for indeterminate matrix entries and polynomial coefficients are nonoptimal when s is fixed and the entries and coefficients are known. Good algorithms for this case are given and tight bounds are derived on the combinational complexity of the most complex matrix-vector function and the most complex polynomial evaluation function. These are operations used in the Deep Space Station computers for decoding telemetry and interpolating ephemerides for antenna pointing and programmed oscillators.

I. Introduction

The two problems treated in this article are the calculations of matrix-vector products and the evaluation of a set of polynomials. The first problem causes most of the combinational complexity in DSN telemetry decoders, and the second tends to overload programmable oscillator minicomputers in high doppler situations such as in planetary orbiters (see Ref. 1 for a data decoder assembly article).

The multiplication of an $m \times n$ matrix A with entries $\{a_{ij}\}$ by an n-vector

$$\mathbf{x} = (x_1, \cdots, x_n)$$

[1]Division of Engineering and Center for Computer and Information Science, Brown University, Providence, Rhode Island, and consultant, Communications Research Section. A portion of the research reported here was completed at Brown University with the support of grants NSF GJ-32 270 and DA-ARO-D-31-124-73-G65.

is defined by

$$f_i(x_1, \cdots, x_n) = a_{i1} * x_1 + \cdots + a_{in} * x_n, \qquad 1 \leq i \leq m \tag{1}$$

where $*$ denotes multiplication and $+$ denotes addition. The evaluation of m polynomials, $b_1(z), \cdots, b_m(z)$ of degree n in one variable z is defined by

$$b_i(z) = b_{i0} + b_{i1} * z + b_{i2} * z^2 + \cdots + b_{in} * z^n, \qquad 1 \leq i \leq m \tag{2}$$

where $*$ and $+$ are the multiplication and addition operators and $z^j = z \cdot z^{j-1}$ denotes the j-fold product of z with itself.

The obvious way to do matrix-vector multiplication is that indicated by Eq. (1) and requires mn $*$ operations and $m(n-1)$ $+$ operations. Polynomials can be evaluated with Horner's rule, namely,

$$b_i(z) = b_{i0} + z(b_{i1} + \cdots) \cdots)$$

which is not so obvious and which uses mn $*$'s and mn $+$'s. It can be shown (Refs. 2, 3, and 4) that no fewer than these numbers of operations are sufficient when the matrix entries, vector entries, polynomial coefficients, and the parameter z are indeterminates (or are isomorphic to indeterminates).

In this article, it is assumed that the vector elements $\{x_1, x_2, \cdots, x_n\}$ and the parameter z are the only indeterminate elements and that the matrix entries $\{a_{ij}\}$ are known, fixed, and drawn from a set of size s, a typical DSN situation. Then the functions $\{f_1, f_2, \cdots, f_m\}$ are functions of n variables and the polynomials $\{b_1(z), \cdots, b_m(z)\}$ are functions of one variable. By exhibiting algorithms, we shall show that the set $\{f_1, \cdots, f_m\}$ can be computed with approximately $mn\,\ell n(s)/\ell n(m)$ operations and the set $\{b_1(z), \cdots, b_m(z)\}$ can be computed with about $mn\,\ell n(s)/\ell n(mn)$ operations, for m and n large. Furthermore, counting arguments will be derived to show that under suitable conditions these upper bounds can be improved upon by at most constant factors for the worst-case matrix and worst-case set of polynomials.

We conclude that algorithms for computing matrix-vector products for fixed matrices and for evaluating a specific set of polynomials are asymptotically less expensive to realize than algorithms which compute any $m \times n$ matrix-vector product or for evaluating any set of m n-degree polynomials. These results also hold when the number of values assumed by matrix elements and by polynomial coefficients is bounded.

II. Preliminaries

Let $\mathbf{f} = \{f_1, \cdots, f_m\}$ be the set of functions realized by the matrix-vector product and let

$$\mathbf{f} = A\mathbf{x} \tag{3}$$

represent the m expressions defined in Eq. (1), where $\mathbf{x} = (x_1, \cdots, x_n)$. Assume that the x_i are indeterminate over the set S so that $\mathbf{f}: S^n \to S^m$, where S^n denotes the n-fold Cartesian product of S with itself. Since the coefficients $\{a_{ij}\}$ in Eq. (1) are assumed drawn from a finite set, without loss of generality, we let a_{ij} be chosen from among $\{1, 2, \cdots, s\}$ and regard the $*$ operation $a_{ij} * x_j$ to be the pth unary operation $U_p(x_j)$, $U_p: S \to S$, when $a_{ij} = p$. We also define $+ : S^2 \to S$ to be an associative binary operation and call it addition. Also, we assume that S contains the additive unit o satisfying $o + x = x + o = x$.

The following examples illustrate the generality of this formulation:

(1) Let $s = 2$, $U_1(x_j) = 0$, $U_2(x_j) = x_j$. Then, the algebraic system $< S, + >$ is a semigroup such as:

  (a) $S = R$ (reals), $+$ denotes either multiplication or addition.

  (b) $S = \{0, 1\}$, $+$ denotes the Boolean AND, OR, or EXCLUSIVE OR.

(2) (a) $S = R$, $U_p(x) = r_p * x$, $r_p \varepsilon R$, $*$ denotes multiplication on reals and $+$ denotes addition on reals.

  (b) $S = D^{q \times q}$ (set of $q \times q$ matrices over D), $U_p(x) = x^p$, the pth power of the matrix x, $+$ denotes matrix addition.

  (c) $S = \{0, 1\}$, $U_1(x) = \bar{x}$ (Boolean complement), $U_2(x) = x$, f_i are called minterms.

The polynomials defined by Eq. (2) are $\{b_1(z), \cdots, b_m(z)\}$ as above. Let $z \varepsilon S$ so that $b_i: S \to S$ and let b_{ij} be in $\{1, 2, \cdots, s\}$, $b_{ij} * z^j$ denote the unary operation $U_p(z^j)$, $U_p: S \to S$, when $b_{ij} = p$. Also, let $\cdot$ and $+$ be binary associate operations, $\cdot : S \times S \to S$, $+ : S \times S \to S$, let $\cdot$ distribute over $+$ and let z^j be defined by $z^j = z \cdot z^{j-1}$. Examples are:

(1) $S = R$ (reals), $U_p(z^j) = r_p * z^j$, $r_p \varepsilon R$, and $* = \cdot$ and is multiplication on reals, and $+$ is real addition.

(2) $S = D^{q\times q}$, $s = 2$, $\cdot$ and $+$ denote matrix product and addition and $U_1(z^j) = 0$, $U_2(z^j) = z^j$.

The complexity of a set of functions $\{g_1, g_2, \cdots, g_m\}$ over S, $g_i : S^{t_i} \to S$, will be measured by the minimum number of steps to realize the set with *straight-line algorithms* (SLAs). The relationships between this measure and conventional measures of complexity such as storage and time are discussed in Ref. 5.

Definition. Let $\Omega = \{h_i \mid h_i : S^{n_i} \to S\}$ be a finite set of functions over S, called the *basis*, and let $\Gamma = \{x_1, \cdots, x_k, K\}$, called the *data set*, be a set of k indeterminates over S and $K \subset S$, a set of constants. Then a q-step SLA, $\beta = (\beta_1, \cdots, \beta_q)$, is an ordered set of q-steps $\{\beta_1, \cdots, \beta_q\}$ where either $\beta_j \varepsilon \Gamma$ (a data step), or

$$\beta_j = (h_i ; \beta_{j_1}, \cdots, \beta_{j_{n_i}}), \qquad j_1, \cdots, j_{n_i} < j$$

that is, β_j is a computation step which results from a basis function operating on previous steps. The SLA computes the set of functions $\{\bar{\beta}_1, \cdots, \bar{\beta}_q\}$ where

$$\bar{\beta}_j = \begin{cases} \beta_j & \text{if } \beta_j \varepsilon \Gamma \\ h_i(\bar{\beta}_{j_1}, \cdots, \bar{\beta}_{j_{n_i}}) & \text{otherwise} \end{cases}$$

Definition. To each basis element h_i, assign cost $c_i > 0$. Then, the cost of an SLA β, $x(\beta)$, is defined by

$$x(\beta) = \sum_{i=1}^{|\Omega|} q_i c_i$$

where q_i is the number of occurrences of h_i in β and $|\Omega|$ is the number of basis functions. The *combinational complexity* of the set of functions $\mathbf{g} = \{g_1, \cdots, g_m\}$, $g_i : S^{t_i} \to S$, $C_\Omega(\mathbf{g})$, is the minimum of $x(\beta)$ over all SLAs β which compute g and undefined (or infinite) if there is no β with basis Ω which computes g.

III. Upper Bounds

Let $C_{m,n}$ denote the maximal combinational complexity of the $m \times n$ matrix-vector product functions and let $D_{m,n}$ be the same for the most complex set of m polynomials of degree n. In this section, upper bounds to $C_{m,n}$ and $D_{m,n}$ are derived by construction of SLAs. Lower bounds to these quantities are derived in the next section.

We sketch SLAs for evaluation of $\mathbf{f} = A\mathbf{x}$ and of $\{b_1(z), \cdots, b_m(z)\}$. We begin with the matrix-vector product problem. Represent the n-vector x as

$$t_0 = \left\lceil \frac{n}{k} \right\rceil$$

($\lceil x \rceil$ is the smallest integer $\geqq x$) subvectors

$$\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^{t_0})$$

where

$$\mathbf{x}^i = (x_{(i-1)k+1}, \cdots, x_{ik})$$

for $1 \leqq i \leqq t_0 - 1$ and

$$\mathbf{x}^{t_0} = (x_{(t_0-1)k+1}, \cdots, x_n)$$

Let A be subdivided into t submatrices $B_1, B_2, \cdots, B_{t_0}$

$$A = [B_1 \quad B_2 \quad \cdots \quad B_t]$$

where B_i is $m \times k$, $1 \leqq i \leqq t_0 - 1$ and B_{t_0} is $m \times r$ where $r = n - (t_0 - 1)k$. Then, $\mathbf{f} = A\mathbf{x}$ can be computed as follows:

(1) Form the t_0 matrix-vector products

$$\mathbf{y}^i = B_i \mathbf{x}^i \qquad 1 \leqq i \leqq t_0 \tag{4}$$

(2) Do vector addition of these products to form f.

$$\mathbf{f} = \mathbf{y}^1 + \cdots + \mathbf{y}^{t_0} \tag{5}$$

This decomposition is possible because the binary operation of addition is associative. Also, we shall show that the products $B_i\mathbf{x}^i$ can be done with significantly fewer operations than are required for the obvious method of matrix-vector multiplication. This will translate into a savings for the computation of f.

Consider polynomial evaluation next. Represent a polynomial $b_i(z)$ as

$$b_i(z) = P_{i1}(z) + P_{i2}(z) \cdot z^k + \cdots + P_{it}(z) \cdot z^{(t_1-1)k} \tag{6}$$

where each $P_{ij}(z)$ is a polynomial of degree $k - 1$ and

$$t_1 = \left\lceil \frac{n+1}{k} \right\rceil$$

This representation is unique and possible because $+$ and $\cdot$ are associative and $\cdot$ distributes over $+$. The algorithm which shall be given for evaluating a set of polynomials will generate all polynomials of degree $k - 1$, use them in the appropriate places, and follow them with the

necessary multiplications by terms z^j and the necessary additions.

Theorem 1. Let Ω contain the binary operation $+$ and the unary operators $\{U_1, \cdots, U_s\}$. If c_a is the cost of $+$, then

$$C_{m,n} \leq \left(\left\lceil \frac{n}{k} \right\rceil - 1\right) C_{m,k} + C_{m,r} + \left(\left\lceil \frac{n}{k} \right\rceil - 1\right) mc_a \tag{7}$$

Proof. In the decomposition of Eqs. (4) and (5),

$$\left(\left\lceil \frac{n}{k} \right\rceil - 1\right) C_{m,k} + C_{m,r}$$

is the cost of computing $\mathbf{y}^1, \cdots, \mathbf{y}^{t_0}$. With a total of

$$\left(\left\lceil \frac{n}{k} \right\rceil - 1\right)$$

additions per component of $\mathbf{f}$, $\mathbf{f}$ can be formed. Since this decomposition may not be the best way to realize $\mathbf{f}$, Eq. (7) follows.

Theorem 2. Let Ω contain the two binary operations $+$ and $\cdot$ and the unary operators $\{U_1, \cdots, U_s\}$. Also, let $U_s(x) = x$ be the identity operator. If c_a is the cost of $+$ and c_b is the cost of $\cdot$, then

$$D_{m,n} \leq D_{N,k-1} + \left(\left\lceil \frac{n+1}{k} \right\rceil - 1\right)(mc_a + (m+1)c_b) \tag{8}$$

where $N = s^k$ is the number of polynomials of degree $k-1$.

Proof. In the decomposition of Eq. (6), all of the polynomials $P_{ij}(z)$ are fixed. There are at most $N = s^k$ of them and they can *all* be realized with a combinational complexity of $D_{N,k-1}$. To form Eq. (6), compute $z^k, \cdots, z^{(t_1-1)k}$. This can be done with $(t_1 - 1)$ $\cdot$'s since z^{k-1} is available as a polynomial of degree $k-1$. Form the $t_1 - 1$ products $P_{ij}(z) \cdot z^j$, $1 \leq j \leq t_1 - 1$, for each $i, 1 \leq i \leq m$ and do the indicated additions. A total of $m(t_1 - 1)$ additions and $(m+1)(t_1 - 1)$ multiplications will be done. Then Eq. (8) follows since this is but one method for computing $\{b_1(z), \cdots, b_m(z)\}$.

These two theorems will prove useful when bounds to $C_{m,k}$ and $D_{N,k-1}$ have been derived. This is the next task.

Lemma 1. Let Ω contain $+$ and $\{U_1, \cdots, U_s\}$.

Let $\varepsilon = 1$ if one of the unary operators, say U_1, is the o-ary operator, $U_1(x_j) = 0$, and let $\varepsilon = 0$, otherwise. Let c_p be the cost of U_p, $1 \leq p \leq 2$, and let

$$K_1 = mk\left(\max_{1 \leq t \leq s} c_t\right) + m(k-1)c_a$$

$$K_2 = \begin{cases} k(c_1 + \cdots + c_s) + c_a(s^{k+1} - s^2)/(s-1) & \varepsilon = 0 \\ k(c_1 + \cdots + c_s - (s-1)c_a) + c_a(s^k - 1) & \varepsilon = 1 \end{cases}$$

then

$$C_{m,k} \leq \min(K_1, K_2)$$

Proof. The K_1 bound follows from the standard aglorithm for matrix-vector multiplication.

The K_2 bound is derived by first noting that all s unary operations on $x_1, \cdots, x_k$ can be performed at a cost of $k(c_1 + c_2 + \cdots + c_s)$ and then determining the number of additions to form all s^k sums of the form

$$U_{t_1}(x_1) + \cdots + U_{t_k}(x_k)$$

If all such sums are available, all functions $\mathbf{f} = A\mathbf{x}$ can be computed since there are at most s^k such distinct functions.

We show by induction that all s^k sums can be formed with at most $s^2 + s^3 + \cdots + s^k = (s^{k+1} - s^2)/(s-1)$ additions. Clearly, all sums in two variables can be formed with s^2 additions. Assume that all s^{k-1} sums in $k-1$ variables $k \geq 3$ can be performed with $s^2 + \cdots + s^{k-1}$ additions. For each of these s^{k-1} sums, s additions are formed to adjoin $\{U_l(x_k), 1 \leq l \leq s\}$, for a total of s^k new additions. Thus, all sums of k variables can be done with $s^2 + s^3 + \cdots + s^k$ additions, and the induction hypothesis holds.

If there is a unary operator $U_1 = 0$, some additions are unnecessary. Form all sums

$$U_{t_i}(x_i) + U_{t_j}(x_j), \qquad l_i \geq 2, \qquad l_j \geq 2, \qquad i \neq j, \qquad 1 \leq i, \qquad j \leq k$$

with one addition per sum. All sums

$$U_{t_i}(x_i) + U_{t_j}(x_j) + U_{t_p}(x_p), \qquad l_i \geq 2, \qquad l_j \geq 2, \qquad l_p \geq 2$$

can be formed with one more addition per sum. Hence, all the sums can be formed with a number of additions

equal to the number of terms requiring one or more additions. Since there is one 0 sum and $k(s-1)$ sums involving one nonzero unary operation, all sums can be formed with $s^k - k(s-1) - 1$ additions when $\varepsilon = 1$.

The next result is a bound on $D_{N,k-1}$.

Lemma 2. Let Ω contain $+$, $\cdot$, and $\{U_1, \cdots, U_s\}$. Let c_a, c_b, and c_p be the costs of $+$, $\cdot$, and U_p, $1 \leq p \leq s$, respectively. Also, assume that $U_s(x) = x$, $c_s = 0$, and let $\varepsilon = 1$ if $U_1(x) = 0$ and let $\varepsilon = 0$ if $U_1(x) \neq 0$. Then, when $N = s^k$,

$$D_{N,k-1} \leq h(k)\, c_b + k(c_1 + c_2 + \cdots + c_{s-1}) + K_2$$

where K_2 is defined above and $h(k) = k - 2$ if $k \geq 2$ and $h(k) = 0$, otherwise.

Proof. $D_{N,k-1}$ is the combinational complexity of the most complex set of s^k polynomials of degree $k-1$. Clearly, this is the set of all polynomials of degree $k-1$.

Realize these polynomials as follows: (1) construct the sequence $1, z, z^2, \cdots, z^{k-1}$ using $h(k)$ $\cdot$'s; (2) to each term apply $U_1, U_2, \cdots, U_{s-1}$ at a cost of $k(c_1 + c_2 + \cdots + c_{s-1})$; (3) then add the k terms. This last step has cost K_2, as shown in the proof of Lemma 1.

These two lemmas and two theorems are combined to give asymptotic bounds on the combinational complexity of matrix-vector multiplication and polynomial evaluation.

Theorem 3. Under the conditions of Theorem 1 and Lemma 1,

$$C_{m,n} \leq \left\lceil \frac{n}{k} \right\rceil (K_2 + mc_a) - mc_a$$

and for m, n large

$$C_{m,n} \leq \frac{mnc_a}{\log_s m}\left(1 + 0\left(\frac{1}{\log_s m}\right) + 0\left(\frac{1}{n}\right)\right)$$

Proof. The first inequality follows directly from Theorem 1 and Lemma 1 since $C_{m,r} \leq C_{m,k}$.

The second inequality follows from a long but uncomplicated set of steps when

$$k = \lceil \log_s [(m/s)/\log_s (m/s)] \rceil$$

Theorem 4. Under the conditions of Theorem 2 and Lemma 2,

$$D_{m,n} \leq \frac{n(mc_a + (m+1)c_b)}{\log_s (mn)}(1 + 0(1/\ln(mn)))$$

for large n where c_a and c_b are the cost of $+$ and $\cdot$, respectively.

Proof. The theorem follows from Theorem 2 and Lemma 2 when

$$k = \lceil \log_s ((mn)/s(\log_s (mn/s))^2) \rceil$$

Table 1 shows the first bound of Theorem 3 to $C_{m,n}$ optimized under variation of k, when $m = n$ is a power of 2, $s = 2$, $c_a = 1$, $c_1 = c_2 = 0$ and $\varepsilon = 1$ (i.e., $U_1(x) = 0$). It also shows the ratio of the K_1 bound to this bound. The improvement over the K_1 bound is substantial and for large binary matrices recommends the method of matrix-vector multiplication by decomposition.

Table 2 shows the bound to D_{1n} when $c_a = c_b = 1$, $c_1 = c_2 = 0$ and $\varepsilon = 1$, which follows from Theorem 2 and Lemma 2. Also shown is the ratio of $2n$, the number of operations for Horner's rule, and this bound. The improvement is substantial for large n and warrants use of the algorithm introduced above.

IV. Lower Bounds

In this section, an upper bound is derived on the number of distinct sets of m functions $\{g_1, \cdots, g_m\}, g_j : S^n \to S$, with $C_\Omega(g_1, \cdots, g_m) \leq C$. If C is not large enough, not all sets of such functions can be realized at a cost $\leq C$. Consequently, at least one set must have combinational complexity $> C$. This result is used to derive lower bounds to $C_{m,n}$ and $D_{m,n}$ and to show that the upper bounds given above are tight under suitable conditions.

Let $N_\Omega(C, m, p)$ be the number of distinct sets of m functions in p variables $\{g_1, \cdots, g_m\}, g_j : S^p \to S$, with $C_\Omega(g_1, \cdots, g_m) \leq C$.

Lemma 3. Let all operations in Ω be either unary or binary. Then, for $p \geq 6$,

$$N_\Omega(C, m, p) < |\Omega|^{C/c^*}(p + |K| + C/c^*)^{2(p+|K|+C/c^*+(m-1)/2)}$$

where $c^* > 0$ is the cost of the minimum cost operation and $K \subset S$ is the set of constants in the data set.

Proof. Consider SLAs in which variable data steps precede constant data steps and these precede computation steps. There is no loss of generality in this assumption.

Let an SLA β have $p' \leq p$ variable data steps $x_{i_1}, \cdots, x_{i_{p'}}$, drawn without replacement from $\{x_1, \cdots, x_p\}$. There are

$$\binom{p}{p'} < 2^p$$

ways to do this.

Let the SLA β have $d \leq |K|$ constant data steps. There are

$$\binom{|K|}{d} < 2^{|K|}$$

ways to choose these d constants.

Also, let β have t computation steps $(h_i; \beta_{i_1}, \beta_{i_2})$ where β_{i_2} may be empty if h_i is unary. There are at most $M(p', d, t)$ such steps, where

$$M(p', d, t) = |\Omega|^t \prod_{j=p'+d}^{p'+d+t} (j-1)^2$$

To β we associate sets of m functions. One function of each set must be associated with the last step (otherwise, the SLA is not optimal) and the remaining $m-1$ functions of each set can be assigned in at most $(p'+d+t-1)^{m-1}$ ways.

There are at most $L(p', d, t)$ sets of m functions associated with SLAs with p' variable data steps, d constant data steps, and t computation steps, and

$$L(p', d, t) \leq 2^{p+|K|} (p'+d+t)^{m-1} M(p', d, t)$$

This bound is clearly monotonically increasing in p', d, and t.

Since $N_\Omega(C, m, p)$ is the sum of $L(p', d, t)$ over $0 \leq p' \leq p$, $0 \leq d \leq |K|$ and $0 \leq t \leq T$ where $T = \lfloor C/c^* \rfloor$ ($\lfloor x \rfloor$ is the largest integer $\leq x$) and $c^* > 0$ is the cost of the minimum cost basis function,

$$N_\Omega(C, m, p) \leq (p+1)(|K|+1)(T+1) 2^{p+|K|} \times (p+|K|+T)^{m-1} M(p, |K|, T)$$

It is easily shown that

$$\sum_{v=L}^{Q} \ln v \leq \int_L^{Q+1} \ln x \, dx = x \ln\left(\frac{x}{e}\right)\Big|_L^{Q+1}, \qquad L \geq 0$$

so that

$$M(p, |K|, T) \leq |\Omega|^T \left(\frac{p+|K|+T}{e}\right)^{2(p+|K|+T)} \times \left(\frac{p+|K|-1}{e}\right)^{-2(p+|K|-1)}$$

Then, combining terms, we have

$$N_\Omega(C, m, p) \leq \left[\frac{(p+1)(T+1)}{e^{2(p+|K|+T)}}\right]\left[\frac{(|K|+1) 2^{p+|K|}}{\left(\frac{p+|K|-1}{e}\right)^{2(p+|K|-1)}}\right] \cdot |\Omega|^T (p+|K|+T)^{2(p+|K|+T+(m-1)/2)}$$

It can be shown that the first bracketed term is monotonically decreasing in T and p and has value ≤ 1 at $T = p = 0$. The second bracketed term is monotonically decreasing in increasing $|K|$ for $p \geq 4$ and is < 1 for $p \geq 6$. From this and $T \leq C/c^*$, the theorem follows.

It remains to apply Lemma 3 in order to derive lower bounds for $C_{m,n}$ and $D_{m,n}$.

Theorem 5. Let Q be the number of distinct sets of m functions $\{g_1, \cdots, g_m\}$, $g_j : S^p \to S$, in the set $\mathcal{G}$. Then, if $C(\mathcal{G})$ is the maximum of $C_\Omega(g_1, \cdots, g_m)$,

$$C(\mathcal{G}) \geq \frac{c^*}{2} \frac{\ln Q}{\ln(\ln Q) + \ln(|\Omega|/2)} - c^*(p + |K| + (m-1)/2)$$

for $p \geq 6$ when $|\Omega| \ln Q \geq 2e$ and $c^* > 0$ is the cost of the minimum cost basis function.

Proof. Let C_0 satisfy

$$2x(\ln x + \ln|\Omega|) = \ln Q$$

where

$$x = p + |K| + (C_0/c^*) + (m-1)/2$$

Then, it is easy to show that $N_\Omega(C_0, m, p) < Q$. Consequently, for some set

$$\{g_1, \cdots, g_m\} \in \mathcal{G}, \qquad C_\Omega(g_1, \cdots, g_m) \geq C_0$$

We solve for C_0. The equation above is rewritten as

$$y \ln y = \frac{|\Omega|}{2} \ln Q = B$$

where $y = x|\Omega|$. The function $y \ln y$ is monotonically increasing for $y \geq 1$ and at $y = y_1 = B/\ln B$ we have

$$y_1 \ln y_1 = \frac{B}{\ln B}(\ln B - \ln \ln B) < B$$

if $\ln \ln B \geq 0$ or $B \geq e$. Therefore, the solution y is

$$y \geq y_1 = \frac{\frac{|\Omega|}{2} \ln Q}{\ln (\ln Q) + \ln (|\Omega|/2)}$$

if $|\Omega| \ln Q \geq 2e$ and it follows from

$$y = (p + |K| + (C_0/c^*) + (m-1)/2|\Omega|$$

that

$$C_0 \geq \frac{c^*}{2} \frac{\ln Q}{\ln (\ln Q) + \ln (|\Omega|/2)} - c^*(p + |K| + (m-1)/2)$$

This completes the proof.

Note that the proof of Theorem 5 and Lemma 3 requires that the cost of all basis functions be greater than zero. Thus, if basis functions of zero cost exist, they cannot be used if these results are to apply. We now specialize these results to matrix-vector multiplication and polynomial evaluation.

Corollary 1. If all of the s^n $1 \times n$ matrix-vector functions are distinct and if there are α_1, α_2 with $0 < \alpha_1 < \alpha_2 < \infty$ such that $\alpha_1 \leq m/n \leq \alpha_2$, then

$$C_{m,n} \geq \frac{c^* mn}{4 \log_s (m)} (1 + 0(1/\ln (m)))$$

for $|\Omega|$, $|K|$, and s fixed and m, n large.

Proof.

$$Q = \binom{s^n}{m}$$

since there are this many ways to construct m distinct functions in $p = n$ variables. Then,

$$Q = s^{nm}(1 - 1/s^n) \cdots (1 - (m-1)/s^n)/m!$$

Using the inequality

$$(1 - a_1)(1 - a_2) \geq 1 - a_1 - a_2$$

for $a_1, a_2 \geq 0$, we have

$$Q \geq s^{nm}(1 - m(m-1)/(2s^n))/m! > s^{nm}(1 - m^2/s^n)/m!$$

Then,

$$\ln Q \geq nm \ln s + \ln (1 - m^2/s^n) - m \ln m$$

or

$$\ln Q \geq nm \ln s \left(1 + 0\left(\frac{m}{ns^n} + \frac{\ln m}{n}\right)\right)$$

since $\ln (1 - x) \leq x$ and $m^2/s^n << 1$ under the conditions stated.

From the monotonicity of $x/(\ln x + a)$ for $\ln x \geq 1 - a$ and from the discussion above, it follows that

$$\frac{\ln Q}{\ln \ln Q + \ln (|\Omega|/2)} \geq \frac{nm}{\log_s (nm)}\left(1 + 0\left(\frac{m}{ns^n} + \frac{\ln m}{n}\right) + 0\left(\frac{\ln |\Omega|}{\ln (mn)}\right)\right)$$

and since $\alpha_1 \leq m/n \leq \alpha_2$, m and n large, the dominant term is the last one. Finally $\log mn \leq 2 \log m - \log \alpha_1$, from which the theorem follows.

Comparing this corollary with the upper bound of Theorem 3, it is clear that when m and n are comparable and large and all the $1 \times n$ matrix-vector functions are distinct, the upper bound to $C_{m,n}$ can be improved by at most a constant factor. Certainly all the $1 \times n$ matrix-vector functions are distinct when $s = 2$ and $U_1(x) = 0$, $U_1(x) = x$, and $x_i \varepsilon S = R$ (reals).

Corollary 2. If all of the s^{n+1} n-degree polynomials in z are distinct and if there is an α with $0 < \alpha < \infty$ such that $m < \alpha n$, then for large n

$$D_{m,n} \geq \frac{c^*}{2} \frac{mn}{\log_s (mn)} (1 + 0(1/\ln (mn)))$$

for $|\Omega|$, $|K|$, and s fixed.

Proof. The number of distinct sets of m functions Q equals

$$\binom{s^{n+1}}{m}$$

Using the lower bound of the proof of Corollary 1 with $n + 1$ replacing n, we have

$$\frac{\ln Q}{\ln \ln Q + \ln(|\Omega|/2)} \geqq \frac{(n+1)m}{\log_s (n+1)m} \times \left(1 + 0\left(\frac{m}{(n+1)s^{n+1}} + \frac{\ln m}{n+1}\right) + 0\left(\frac{1}{\ln (n+1)m}\right)\right)$$

Under the condition $m \leqq \alpha n$, $0 < \alpha < \infty$, the dominant term for large n is the last term.

Since the polynomials depend on $p = 1$ variable, from Theorem 5 we have

$$D_{m,n} \geqq \frac{c^*}{2} \frac{(n+1)m}{\log_s (n+1)m} (1 + 0(1/\ln(nm))) - c^*(1 + |K| + (m-1)/2)$$

For n large, the theorem follows from the suitable approximation to this bound.

The result for polynomial evaluation is somewhat stronger than that for matrix multiplication since the bounds of Corollary 2 and Theorem 4 differ by at most a constant for large n only. It is not necessary that m, the number of polynomials, also be large. Hence, when $m = 1$ and one polynomial of degree n is to be evaluated, the combinational complexity of the worst such function behaves as $n/\log_s n$, for n large, for those cases where there are s^n distinct n-degree polynomials. One such case is that in which $s = 2$, $U_1(z^j) = 0$, $U_2(x^j) = z^j$ and $z \varepsilon R$.

V. Conclusions

The matrix-vector functions and the polynomial functions examined here are restrictions of the functions which obtain when the matrix elements and the polynomial coefficients are treated as indeterminates. From this vantage point, it is not surprising that these two problems are considerably less complex than the general problems. Nevertheless, the algorithms presented here promise considerable reductions in the number of operations to do matrix-vector multiplication with fixed matrices and to do polynomial evaluation with fixed polynomials. These reductions, however, will be realized only in those applications where the matrix-vector multiplication and polynomial evaluation are to be done many times, since the algorithms offered above must be constructed from a search of the matrix entries and of the polynomial coefficients and this search time will be comparable to the time to evaluate the functions using the algorithms for the general problem.

References

1. Grauling, C. R., and Jones, N. J., "Performance Capabilities of the Data Decoder Assembly Through the Viking Era," in *The Deep Space Network Progress Report*, Technical Report 32-1526, Vol. X, pp. 164–168, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 15, 1972.
2. Pan, V. Ya., "Methods of Computing Values of Polynomials," *Russian Mathematical Surveys*, Vol. 21, No. 1, Jan.–Feb. 1966, pp. 105–136.
3. Winograd, S., "On the Number of Multiplications Necessary to Compute Certain Functions," *Communications on Pure and Applied Mathematics*, Vol. 23, 1970, pp. 165–179.
4. Winograd, S., *On the Algebraic Complexity of Functions*, IBM Report, International Business Machines Corp., Armonk, N.Y., Nov. 1968.
5. Savage, J. E., "Computational Work and Time on Finite Machines," *Journal of the Association for Computing Machinery*, Vol. 19, No. 4, Oct. 1972, pp. 660–674.

Table 1. Bound on $C_{n,n}$

n	Best k	$C_{n,n} \leq$	$n(n-1)$	Ratio
4	2	6	12	2
8	2 or 3	28	56	2
16	4	92	240	2.61
32	4	312	992	3.18
64	5	1,106	4,032	3.65
128	5	3,876	16,256	4.19
256	6	13,203	65,280	4.94
512	7	46,256	261,632	5.66
1024	8	161,664	1,047,552	6.48
2048	9	585,472	4,192,256	7.16
4096	10	2,090,594	16,773,120	8.02

Table 2. Bound on $D_{1,n}$

n	Best k	$D_{1,n} \leq$	$2n$	Ratio
3	2	4	6	1.5
7	2	10	14	1.4
15	3	20	30	1.5
31	4	34	62	1.82
63	4	58	126	2.17
127	5	104	254	2.44
255	5	182	510	2.80
511	6	322	1022	3.17
1023	7	563	2046	3.63
2047	7	1001	4094	4.09
4095	8	1786	8190	4.59